CLAIMS

What is claimed is:

1. A system for designing electronic circuits, comprising:

a memory device for storing a program;

a processor in communication with the memory device, the processor operative with the program to:

receive a circuit model, wherein the circuit model has one or more circuit gates;

receive a library having one or more logic gates, wherein each logic gate has a topology;

calculate leakage sensitivities for each of the topologies; and

synthesize a new circuit model by selecting one or more of the topologies based on its leakage sensitivities, wherein the new circuit model has reduced current leakage.

2. The system of claim 1, wherein the processor is further operative with the program to:

output the new circuit model.

3. The system of claim 1, wherein the leakage sensitivities are determined by a topology dependent leakage analytical model.

4. The system of claim 1, wherein the leakage sensitivities are determined by measuring the leakage current of the logic gate in a circuit simulator while applying various input patterns to the logic gate.

5. The system of claim 1, wherein the processor is further operative with the program to:

optimize current leakage of the new circuit model.

6. The system of claim 1, wherein the processor is further operative with the program to:

optimize timing of the new circuit model.

7. The system of claim 1, wherein the processor is further operative with the program to:

optimize area of the new circuit model.

8. A method for designing electronic circuits, comprising:

receiving a circuit model, wherein the circuit model has one or more circuit gates;

receiving a library having one or more logic gates, wherein each logic gate has a topology;

calculating leakage sensitivities for each of the topologies; and

synthesizing a new circuit model by selecting one or more of the topologies based on its leakage sensitivities, wherein the new circuit model has reduced current leakage.

9. The method of claim 8, further comprising:

outputting the new circuit model.

10. The method of claim 8, further comprising:

receiving a probability model indicating the probability of one or more of the logic gates being in an input state.

11. The method of claim 8, further comprising:

receiving a probability model indicating the probability of one or more transistors being in an input state.

12. The method of claim 8, wherein the leakage sensitivities are determined by a topology dependent leakage model.

13. The method of claim 8, wherein the synthesizing step further comprises:

adding one or more logic gates to the circuit model to create the new circuit model.

14. The method of claim 8, wherein the synthesizing step further comprises:

deleting one or more logic gates from the circuit model to create the new circuit model.

15. The method of claim 8, wherein the synthesizing step further comprises:

substituting one or more logic gates from the circuit model to create the new circuit model.

16. The method of claim 8, wherein the synthesizing step further comprises:

optimizing current leakage of the new circuit model.

17. The method of claim 8, wherein the synthesizing step further comprises:

optimizing timing of the new circuit model.

18. The method of claim 8, wherein the synthesizing step further comprises:

optimizing area of the new circuit model.

19. The method of claim 8, wherein the logic gates are selected from the group consisting of: and, or, nand, nor, xor, and invert.

20. The method of claim 8, wherein the synthesizing step further comprises:

factoring one or more kernels of the circuit model;

decomposing one or more portions of the circuit model into one or more sub-circuits;

mapping one or more circuit gates in one or more of the sub-circuits to one or more of the logic gates in the library that are substituted for the respective circuit gate; and

modifying one or more of the sub-circuits with one or more buffers to create a buffer tree topology.

21. The method of claim 20, wherein the buffer tree topology comprises buffers having reduced transistor widths.

22. The method of claim 20, wherein the synthesizing step is performed on sub-circuits that do not have critical timing constraints.

23. The method of claim 20, wherein the synthesizing step is performed on sub-circuits that have critical timing constraints.

24. The method of claim 20, wherein the factoring step further comprises:

preventing the sharing of logic terms that create excess buffering; and

minimizing total physical area of the circuits on the chip.

25. The method of claim 20, wherein the mapping step further comprises:

analyzing a tradeoff between gate cloning and buffer insertion.

26. The method of claim 20, wherein the mapping step further comprises:

analyzing topologies comprising one or more transistor stacks.

27. The method of claim 20, wherein the mapping step further comprises:

analyzing a leakage tradeoff by considering the probability that one or more of the logic gates is in an input state.

28. The method of claim 20, wherein the mapping step further comprises:

analyzing a leakage tradeoff by considering the probability that a transistor is in an input state.

29. A computer program product comprising a computer useable medium having computer program logic recorded thereon for designing electronic circuits, the computer program logic comprising:

program code for receiving a circuit model, wherein the circuit model has one or more circuit gates;

program code for receiving a library having one or more logic gates, wherein each logic gate has a topology;

program code for calculating leakage sensitivities for each of the topologies; and

program code for synthesizing a new circuit model by selecting one or more of the topologies based on its leakage sensitivities, wherein the new circuit model has reduced current leakage.

30. The computer program product of claim 29, further comprising:

program code for outputting the new circuit model.

31. The computer program product of claim 29, further comprising:

program code for optimizing current leakage of the new circuit model.

32. A system for designing electronic circuits for reducing leakage power, the system comprising:

an input for inputting a circuit design, wherein the circuit design is input using a hardware description language (HDL) that is capable of providing leakage sensitivities to a cell library;

a compiler for compiling the HDL into machine readable language;

a technology independent optimizer for performing technology independent optimization of the circuit design;

a technology mapper for generating a hardware unit to be associated with the circuit design after the circuit design is processed through the technology independent optimizer;

a leakage optimizer for optimizing a leakage current of the hardware unit by applying a set of leakage sensitivities in the cell library to the hardware unit;

a physical circuit generator for generating a new circuit design according to the hardware unit optimized by the leakage optimizer; and

an output for outputting the new circuit design.

33. The system of claim 32, further comprising:

a timing optimizer for optimizing the timing of the hardware unit.

34. The system of claim 32, further comprising:

an area optimizer for optimizing the area of the hardware unit.

35. The system of claim 32, wherein the leakage sensitivities are stored in the cell library.

36. The system of claim 32, wherein the leakage sensitivities are determined by a topology dependent leakage model.

37. The system of claim 32, wherein the cell library comprises logic gates that are selected from the group consisting of: and, or, nand, nor, xor, and invert.

38. A system for topology selection to minimize leakage power during synthesis, comprising:

a memory device for storing a program;

a processor in communication with the memory device, the processor operative with the program to:

receive a circuit design; and

synthesize a new circuit design having a minimized leakage power, wherein the power of the new circuit is minimized by applying an area or timing optimization algorithm that incorporates leakage sensitivities of device topologies.

39. The system of claim 38, wherein the optimization algorithm is selected from the group consisting of: kernel factoring, decomposition, technology mapping and buffering.

40. The system of claim 38, wherein the device topologies are selected from the group consisting of: and, or, nand, nor, xor, and invert.